Perancangan Keamanan Data pada RESTFul *WebService* dengan Algoritma *Vigenere*

Artikel Ilmiah

Peneliti:
Lanang Pandu Aryoko(672010258)
Indrastanti Ratna Widiasari, M.T.

Program Studi Teknik Informatika
Fakultas Teknologi Informasi
Universitas Kristen Satya Wacana
Salatiga
2016

Perancangan Keamanan Data pada RESTFul *WebService* dengan Algoritma *Vigenere*

Artikel Ilmiah

Diajukan kepada
Fakultas Teknologi Informasi
untuk memperoleh gelar Sarjana Komputer

Peneliti:
Lanang Pandu Aryoko(672010258)
Indrastanti Ratna Widiasari, M.T.

Program Studi Teknik Informatika
Fakultas Teknologi Informasi
Universitas Kristen Satya Wacana
Salatiga
2016

PERPUSTAKAAN UNIVERSITAS
UNIVERSITAS KRISTEN SATYA WACANA
Jl. Diponegoro 52 – 60 Salatiga 50711
Jawa Tengah, Indonesia
Telp. 0298 – 321212, Fax. 0298 321433
Email: library@adm.uksw.edu ; http://library.uksw.edu

PERNYATAAN TIDAK PLAGIAT

Saya yang bertanda tangan di bawah ini:

Nama : Lanang Pandu Aryoko

NIM : 672010258 Email : lngpandu@gmail.com

Fakultas : Teknologi Informasi Program Studi : Teknik Informatika

Judul tugas akhir : Perancangan Keamanan Data Pada RESTFul Web Service dengan Algoritma Vigenere

Pembimbing : 1. Indrastanti R. Widiasari, M.T.

2. _____

Dengan ini menyatakan bahwa:

1. Hasil karya yang saya serahkan ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar kesarjanaan baik di Universitas Kristen Satya Wacana maupun di institusi pendidikan lainnya.
2. Hasil karya saya ini bukan saduran/terjemahan melainkan merupakan gagasan, rumusan, dan hasil pelaksanaan penelitian/implementasi saya sendiri, tanpa bantuan pihak lain, kecuali arahan pembimbing akademik dan narasumber penelitian.
3. Hasil karya saya ini merupakan hasil revisi terakhir setelah diujikan yang telah diketahui dan disetujui oleh pembimbing.
4. Dalam karya saya ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali yang digunakan sebagai acuan dalam naskah dengan menyebutkan nama pengarang dan dicantumkan dalam daftar pustaka.

Pernyataan ini saya buat dengan sesungguhnya. Apabila di kemudian hari terbukti ada penyimpangan dan ketidakbenaran dalam pernyataan ini maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya saya ini, serta sanksi lain yang sesuai dengan ketentuan yang berlaku di Universitas Kristen Satya Wacana.

Salatiga, 16 Februari 2017

Lanang Pandu Aryoko

Tanda tangan & nama terang mahasiswa

PERPUSTAKAAN UNIVERSITAS
UNIVERSITAS KRISTEN SATYA WACANA
Jl. Diponegoro 52 - 60 Salatiga 50711
Jawa Tengah, Indonesia
Telp. 0298 - 321212, Fax. 0298 321433
Email: library@adm.uksw.edu ; http://library.uksw.edu

PERNYATAAN PERSETUJUAN AKSES

Saya yang bertanda tangan di bawah ini:

Nama : Lanang Pandu Aryoko
NIM : 672016258 Email : Lngpandu@gmail.com
Fakultas : Teknologi Informasi Program Studi : Teknik Informatika
Judul tugas akhir : Perancangan Keamanan Data Pada RESTFul Web Service Dengan Algoritma Vigenere

Dengan ini saya menyerahkan hak *non-eksklusif** kepada Perpustakaan Universitas – Universitas Kristen Satya Wacana untuk menyimpan, mengatur akses serta melakukan pengelolaan terhadap karya saya ini dengan mengacu pada ketentuan akses tugas akhir elektronik sebagai berikut (beri tanda pada kotak yang sesuai):

☑ a. Saya mengijinkan karya tersebut diunggah ke dalam aplikasi Repositori Perpustakaan Universitas, dan/atau portal GARUDA

☐ b. Saya tidak mengijinkan karya tersebut diunggah ke dalam aplikasi Repositori Perpustakaan Universitas, dan/atau portal GARUDA**

* *Hak yang tidak terbatas hanya bagi satu pihak saja. Pengajar, peneliti, dan mahasiswa yang menyerahkan hak non-ekslusif kepada Repositori Perpustakaan Universitas saat mengumpulkan hasil karya mereka masih memiliki hak copyright atas karya tersebut.*

** *Hanya akan menampilkan halaman judul dan abstrak. Pilihan ini harus dilampiri dengan penjelasan/ alasan tertulis dari pembimbing TA dan diketahui oleh pimpinan fakultas (dekan/kaprodi).*

Demikian pernyataan ini saya buat dengan sebenarnya.

Salatiga, 16 Februari 2017

Lanang Pandu Aryoko
Tanda tangan & nama terang mahasiswa

Mengetahui,

Indrastanti R. W
Tanda tangan & nama terang pembimbing I

Tanda tangan & nama terang pembimbing II

F-LIB-081

Perancangan Keamanan Data pada RESTFul Web Service dengan Algoritma Vigenere

Oleh,

Lanang Pandu Aryoko
NIM :672010258

ARTIKEL ILMIAH

Diajukan Kepada Program Studi Teknik Informasi guna memenuhi sebagian dari persyaratan untuk mencapai gelar Sarjana Komputer

Disetujui oleh,

Indrastanti Ratna Widiasari, M.T.
Pembimbing 1

Diketahui oleh,

Dr. Dhanapatra T. Palekahelu, M.Pd.	Dr. Kristoko Dwi Hartomo, M.Kom.
Dekan	Ketua Program Studi

FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN SATYA WACANA
SALATIGA
2017

Lembar Pengesahan

Judul Tugas Akhir	: Perancangan Keamanan Data pada RESTFul Web Service dengan Algoritma Vigenere
Nama Mahasiswa	: Lanang Pandu Aryoko
NIM	: 672010258
Program Studi	: Teknik Informatika
Fakultas	: Teknologi Informasi

Menyetujui,

Indrastanti Ratna Widiasari, M.T.
Pembimbing 1

Mengesahkan,

Dr. Dharmaputra T. Palekahelu, M.Pd.
Dekan

Dr. Kristoko Dwi Hartomo, M.Kom.
Ketua Program Studi

Dinyatakan Lulus tanggal: 16 Januari 2017

Reviewer :

- Prof. Dr. Wiranto H. Utomo, M.Kom.

FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN SATYA WACANA
Jalan Diponegoro 52 – 60
Phone. (0298) 321212 (Hunting)
Fax. (0298) 321433
E-mail: fti@uksw.edu
Salatiga 50711 – INDONESIA

LEMBAR PERSETUJUAN PUBLISH JURNAL

Dengan mempertimbangkan isi dari jurnal mahasiswa :

Nama Mahasiswa : Lanang Panda Ampoko
NIM : 672010258

Maka jurnal ini dinyatakan :

LAYAK TERBIT / ~~TIDAK LAYAK TERBIT~~

Menyetujui,

Indrastanti R.W.
Pembimbing 1

Pembimbing 2

Mengetahui,

Prof. Dr. Wiranto H. Utomo, M.Kom.
Reviewer

1. Pendahuluan

REST WebService semakin popular digunakan sebagai media pertukaran data antara perangkat *mobile* dengan *server*[1]. Perangkat *mobile* seperti ponsel berbasis Android sampai bahkan tablet berbasis Windows menggunakan paket data internet yang berbayar atau WiFi untuk dapat mengakses layanan dari *server*. Semakin besar data yang ditransmisikan, semakin lama proses untuk mengirim dan menerima. Hal ini juga akan mengakibatkan jaringan menjadi lebih sibuk, dan juga banyaknya biaya pulsa yang dikeluarkan, jika paket data internet digunakan.

Masalah yang menjadi perhatian pada penelitian ini adalah, *REST WebService* berjalan pada protokol HTTP, yang berarti data yang dikirimkan berupa teks. Protokol HTTP tidak menyediakan fitur untuk mengenkripsi data yang ditransmisikan, dengan kata lain informasi yang keluar-masuk melalui protokol ini, dapat disadap, misalnya dengan aplikasi analisis paket internet WireShark. Jika data yang dilewatkan berupa data vital, seperti contohnya password, maka kebocoran informasi ini dapat merugikan pihak yang menggunakan *REST WebService* tersebut

Untuk mengamankan data yang ditransmisikan, dapat digunakan protokol HTTPS yang didalamnya terdapat fitur enkripsi.Kelemahan pada protokol HTTPS adalah sulit untuk diimplementasikan ketika diakses dengan menggunakan aplikasi mobile (Android, iOS, Windows Mobile).

Pada penelitian ini dirancang kriptografi pada *REST WebService*. Tujuan dari pemodelan ini adalah untuk menyandikan informasi yang dikirim dan diterima antara aplikasi *client* dengan *REST WebService*. Algoritma kriptografi yang dipilih untuk digunakan pada pemodelan ini adalah *Vigenerecipher*. Algoritma ini mudah diimplementaskan pada berbagai bahasa pemrograman dan juga berbagai platform (*web*, mobile, desktop).

2. Tinjauan Pustaka

Penelitian tentang pemanfaatan *REST WebService* yang telah dilakukan sebelumnya, salah satunya adalah penelitian oleh Kurniawan [2]. Pada penelitian tersebut menyebutkan bahwa peran tenaga penjual pada sebuah perusahaan sangatlah vital, karena mereka adalah ujung tombak dalam penjualan produk, karena itu perusahaan membutuhkan sistem yang dapat memantau aktivitas dan mempercepat proses pemesanan produk. Aplikasi berbasis mobile dapat menjadi solusi untuk menyelesaikan masalah tersebut.Aplikasi mobile yang dibuat memanfaatkan data dari GPS untuk memastikan lokasi dari tenaga penjual. Aplikasi yang dibuat juga memiliki fasilitas untuk membaca barcode barang menggunakan kamera untuk mempercepat input data barang. Aplikasi ini menggunakan *REST* Services untuk memanipulasi data yang ada pada layanan komputasi awan.Dengan menggunakan aplikasi mobile ini perusahaan dapat dengan mudah memantau tenaga penjual dan melakukan pemesanan barang dengan lebih cepat dan efisien.

Selain penelitian Kurniawan tersebut, terdapat penelitian yang dilakukan oleh Rahman. Pada penelitian tersebut, diimplementasikan *REST WebService*

untuk game pada perangkat Android [3]. Aplikasi pada perangkat Android, berkomunikasi satu dengan yang lain, melalui data yang dikirim dan diterima pada *server*. Komunikasi aplikasi Android dengan *server* database ini dijembati dengan *REST WebService*. Data yang dikirimkan berada dalam format JSON.

Berdasarkan latar belakang masalah keamanan data mobile dan *webservice*, dan berdasarkan penelitian-penelitian yang telah dilakukan tentang kriptografi maka dilakukan penelitian dengan rumusan masalah yaitu bagaimana mengamankan komunikasi antara *REST WebService*dangan aplikasi *client* yang mengakses data di layanan tersebut. Bagaimana merancang keamanan data dengan Algoritma Vigenere, dan diintegrasikan pada *REST WebService* dan aplikasi *client*.

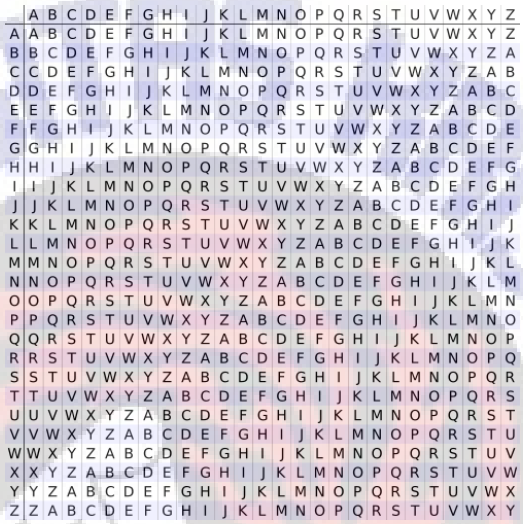
Tujuan dari penelitian ini adalah merancang keamanan data pada *REST WebService* dengan Algoritma Vigenere.Manfaat dari penelitian ini adalah menyediakan perangkat lunak untuk pengamanan data terutama pada layanan *REST WebService*.

Penelitian ini dibatasi pada beberapa hal yaitu: (1) *REST WebService* dikembangkan dengan menggunakan teknologi PHP; (2) Aplikasi mobile yang digunakan untuk uji pengaksesan *REST WebService* terdiri dari aplikasi berbasis Android, aplikasi berbasis desktop dan aplikasi *web* java script; (3) Algoritma kriptografi yang digunakan adalah *Vigenere*.

Penelitian ini merancang sebuah mekanisme untuk pengamanan data pada *REST WebService*. Salah satu teknik pengamanan data adalah kriptografi. Kriptografi (*cryptography*) merupakan ilmu dan seni penyimpanan pesan, data, atau informasi secara aman. Kriptografi (*Cryptography*) berasal dari bahasa Yunani yaitu dari kata *Crypto* dan *Graphia* yang berarti penulisan rahasia[4]. Kriptografi merupakan bagian dari suatu cabang ilmu matematika yang disebut *Cryptology*. Dalam mengenkripsi dan mendekripsi data, kriptografi membutuhkan suatu algoritma (*cipher*) dan kunci (*key*). *Cipher* adalah fungsi matematika yang digunakan untuk mengenkripsi dan mendekripsi. Sedangkan kunci merupakan sederetan bit yang diperlukan untuk mengenkripsi dan mendekripsi data[5]. Secara umum proses kriptografi dibagi menjadi dua bagian yaitu enkripsi dan dekripsi. Data yang telah dienkripsi disebut *ciphertext*karena data asli telah mengalami proses di dalam sebuah algoritma kriptografi atau lebih dikenal dengan nama*cipher*. Kebalikannya, proses mengubah pesan yang telah dienkripsi (*ciphertext*) menjadi pesan asli (*plaintext*) disebut sebagai proses dekripsi.

Algoritma kriptografi yang digunakan pada penelitian ini adalah *VigenereCipher*, *VigenereCipher* merupakan algoritma kriptografi klasik. Operasi pada algoritma kriptografi klasik berbasis pada operasi karakter, sedangkan operasi pada algoritma kriptografi modern berbasis pada operasi *bit*. Dalam kriptografi klasik, *VigenereCipher* termasuk ke dalam *cipher* substitusi abjad majemuk, yang terbuat dari sejumlah *cipher* abjad tunggal, masing-masing dengan kunci yang berbeda[6]. *VigenereCipher* telah berkali-kali diciptakan ulang dengan cukup bervariasi.Namun, metode aslinya digambarkan oleh Giovan Batista Belaso pada tahun 1553 seperti tertulis di dalam bukunya LaCifradel Sig. Giovan Batista Belaso.Meskipun demikian, *VigenereCipher* dipopulerkan oleh Blaise de *Vigenere* pada tahun 1586.*VigenereCipher* menggunakan Bujur Sangkar *Vigenere*

(Gambar 1) untuk melakukan enkripsi.Pada bujur sangkar tersebut, kolom paling kiri menyatakan huruf-huruf kunci, dan baris paling atas menyatakan *plaintext* sedangkan karakter-karakter lainnya menunjukkan karakter *ciphertext*.Karakter *ciphertext* ditentukan dengan menggunakan prinsip *CaesarCipher*. Pergeseran huruf menjadi *ciphertext* ditentukan oleh nilai desimal dari huruf kunci yang bersangkutan (a = 0, b = 1, ..., y = 24, z = 25). *VigenereCipher* menggunakan tabel *Vigenere* yang dikenal dengan *tabularecta* (Gambar 1).



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Gambar1 Tabula *Recta* yang Digunakan oleh *VigenereCipher*[7]

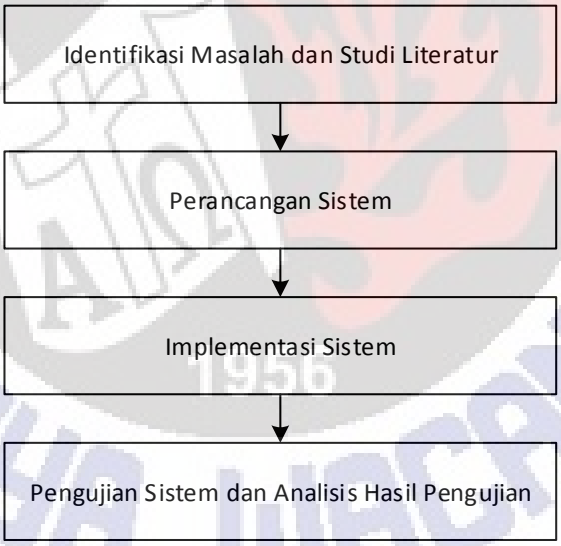
Tabula recta digunakan untuk mendapatkan *ciphertext* dengan menggunakan kunci yang telah ditentukan.Jika panjang kunci lebih pendek daripada panjang *plaintext*, maka kunci diulang penggunaannya (sistem periodik). Jika panjang kunci adalah m, maka periodenya adalah m. Secara singkat, enkripsi dapat digambarkan sebagai berikut: p (*plaintext*): KRIPTOGRAFI k (kunci): LAMPIONLAMP c (*ciphertext*): VRUEBCTCARX

REST(Representational State Transfer) WebService, yang merupakan objek dari penelitian ini,adalah suatu gaya arsitektur perangkat lunak untuk pendistribusian sistem hipermedia seperti WWW. Secara spesifik, *REST* merujuk pada suatu prinsip-prinsip arsitektur jaringan yang menggariskan pendefinisian dan pengalamatan sumber daya. Istilah ini sering digunakan untuk mendeskripsikan semua *interface* sederhana yang mengirimkan data melalui HTTP tanpa ada tambahan lapisan pesan seperti SOAP. Keuntungan lain dari antarmuka *REST* adalah *request* dan *respon* dapat dipendekkan. Prinsip dasar desain *REST* adalah membuat pemetaan *one-to-one* antara operasi *create, read, update, dan delete* yang menggunakan *method* sebagai *POST* untuk membuat sebuah *resource* pada *server.GET*untuk menerima sebuah *resource.PUT* untuk proses *update state* dari *resource. DELETE* untuk menghapus *resource.* Dalam konsep arsitektur *REST WebService*, membuat panggilan ke suatu HTTP API secara signifikan lebih mudah daripada ke SOAP API, karena membutuhkan *libraryclient*, membutuhkan pengenalan, dan kebiasaan. Sedangkan HTTP API

adalah asli dari semua bahasa pemrograman dan hanya melibatkan HTTP *request* dengan parameter sesuai yang ditambahkan, sehingga lebih memudahkan dalam melakukan proses pemanggilan. HTTP API mudah untuk *testing* dan *troubleshoot,* karena dapat membangun panggilan dengan tidak lebih dari sekedar *browsing* dan memeriksa respon dalam jendela *browser* itu sendiri. Karena berbasis HTTP/RESTful, API dapat dikonsumsi menggunakan *request* GET sederhana, dan *server proxy/reverse-proxy* dapat melakukan *cache* atas respon tersebut dengan mudah.Untuk mengakses RESTful *webservice*digunakan sebuah URI (*Uniform Resource Identifiers*) yang merupakan nama dan alamat dari sebuah *resource*. RESTful *webservice* tidak menggunakan WSDL.Pesan yang dikirim, dikemas dalam format XML dan JSON. Berbeda dengan *SOAPwebservice*yang menggunakan protokol khusus untuk pengiriman pesan[8][9]

3. Metode dan Perancangan Sistem

Penelitian yang dilakukan, diselesaikan melalui tahapan penelitian yang terbagi dalam empat tahapan, yaitu: (1) Identifikasi masalah dan studi literatur, (2) Perancangan sistem, (3) Implementasi sistem yaitu perancangan aplikasi/program, dan (4) Pengujian sistem serta analisis hasil pengujian.



Gambar2 Tahapan Penelitian[10]

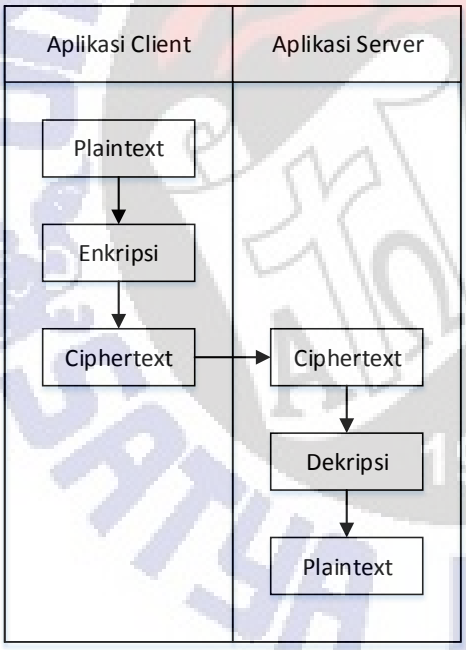
Tahapan penelitian dilakukan dengan mengadaptasi metodologi *waterfall model*[10]. Pada penelitian ini tidak dilakukan dua tahapan terakhir dari *waterfall model*, yaitu*Deployment* dan *Maintenance*, karena pada penelitian ini merupakan sebuah penelitian perancangan.Tahapan penelitian pada Gambar 2, dapat dijelaskan sebagai berikut.*Tahap pertama*: identifikasi masalah, yaitu masalah kemanan data yang ditransmisikan antara *RESTWebServer* dengan *client* yang berupa perangkat *mobile*. Studi literatur dilakukan untuk mencari metode kriptografi data yang dapat diterapkan, serta penelitian-penelitian terdahulu yang dapat digunakan sebagai acuan.

Tahap kedua: perancangan sistem yang meliputi perancangan proses kompresi dan proses dekompresi.Perancangan kedua proses tersebut juga dilakukan pada sisi *client*. Algoritma kriptografi yang digunakan adalah *Vigenere*.

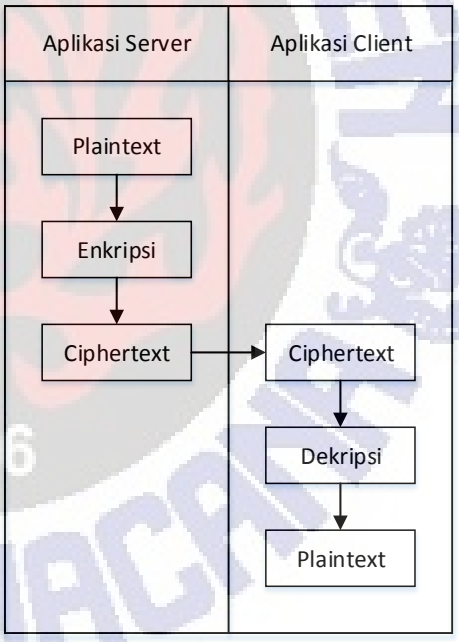
Tahap ketiga: implementasi sistem, yaitu membuat aplikasi sesuai perancangan proses pada tahap kedua. Sistem yang dibuat terdiri dari dua aplikasi, yaitu *server* dan *client*.Aplikasi yang dikembangkan bertujuan untuk mensimulasikan hasil kompresi yang dikirim dan diterima oleh kedua sisi.

Tahap keempat: pengujian sistem dan analisis hasil pengujian, yaitu dilakukan pengujian terhadap proses yang telah dirancang. Pengujian memiliki tujuan utama yaitu melihat keuntungan dan kerugian dengan adanya implementasi kompresi pada komunikasi *REST WebService* dengan *client*.

Sistem yang dikembangkan terdiri dari dua aplikasi.Aplikasi pertama adalah aplikasi *server* lebih tepatnya *REST WebService*.Aplikasi *server* berfungsi untuk menyediakan layanan data.Aplikasi kedua adalah aplikasi *client*.Aplikasi *client* adalah aplikasi yang mengakses dan memanfaatkan layanan data yang disediakan oleh *REST WebService*. Rancangan proses komunikasi dua aplikasi ini ditunjukkan pada Gambar 3 dan Gambar 4.



Gambar 3 Desain Sistem Komunikasi dari *Client* ke *Server*[1]



Gambar 4 Desain Sistem Komunikasi dari *Server* ke *Client*[1]

Gambar 3 merupakan rancangan proses komunikasi antara aplikasi *client* dengan aplikasi *server* dalam hal ini *REST WebService*. *Client* bermaksud mengirimkan data ke *server* dalam bentuk *plaintext*.Sebelum dikirimkan, data yang berupa *plaintext* dienkripsi dahulu dengan algoritma Vigenere, kemudian hasil enkripsi berupa *ciphertext*, dikirimkan ke *server*.*Server* menerima data dalam bentuk *ciphertext*, kemudian didekripsi dengan algoritma Vigenere, sehingga diperoleh data *plaintext*. Selanjutnya data *plaintext* tersebut diteruskan ke proses selanjutnya (disimpan/diolah).

Komunikasi dari *server* ke *client* (Gambar 4) merupakan rancangan komunikasi sebaliknya.*Server* mengirimkan data yang telah terenkripsi dengan algoritma Vigenere sebelumnya (*ciphertext*) ke aplikasi *client*. Kemudian oleh *client*, data *ciphertext* didekripsi dengan algoritma Vigenere, untuk mendapatkan data *plaintext*.Selanjutnya data plaintext tersebut diolah atau ditampilkan di aplikasi *client*.

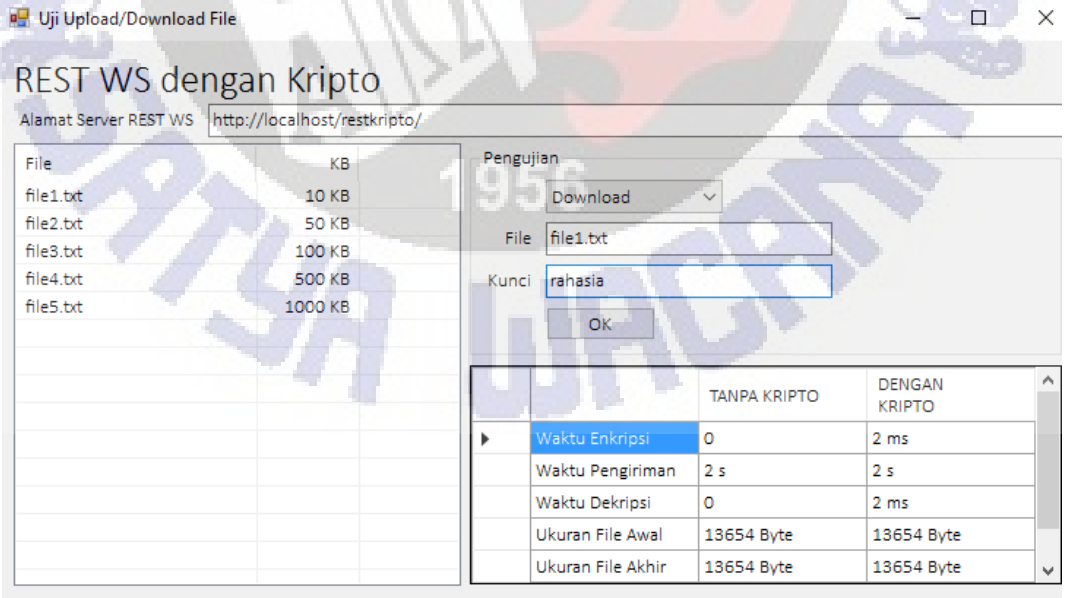
4. Hasil dan Pembahasan

Aplikasi yang dikembangkan sebagai pemodelan kriptografi pada *REST WebService* pada penelitian ini terdiri dari dua bagian. Bagian pertama yaitu aplikasi *server*. Aplikasi *server* tidak memiliki tampilan, karena berfungsi penyedia data. Aplikasi *server* memiliki 4 operasi, ditunjukkan pada Tabel 1.

Tabel 1 Operasi pada *RESTWebServer*

No	Operasi	Keterangan
1	`down`*`Plaintext`*`(filename)`	Fungsi pada *server* yang digunakan oleh *client* untuk mengambil *fileplaintext*
2	`downCipherText(filename)`	Fungsi pada *server* yang digunakan oleh *client* untuk mengambil *fileciphertext*
3	`up`*`Plaintext`*`(filename)`	Fungsi pada *server* yang digunakan oleh *client* untuk mengirim *fileplaintext*
4	`upCipherText(filename)`	Fungsi pada *server* yang digunakan oleh *client* untuk mengirim *fileciphertext*

Bagian yang kedua, berupa aplikasi *client*, memiliki tampilkan yang ditunjukkan pada Gambar 5. Aplikasi *client* pada penelitian ini dirancang sebagai pengujian model enkripsi/dekripsi antara *client* dengan *server* dan sebaliknya.

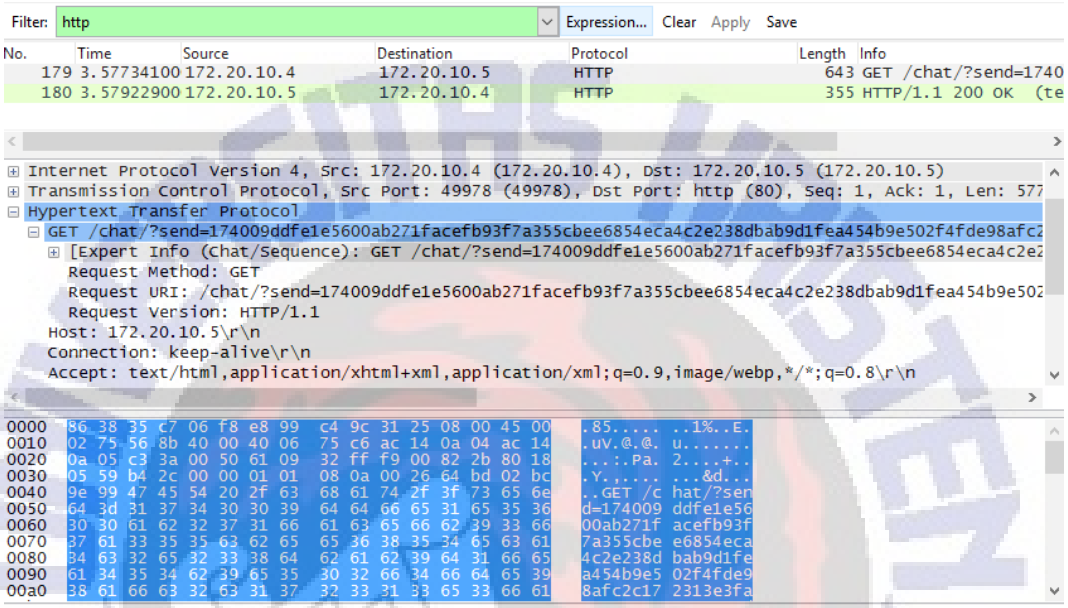


Gambar 5 Tampilan aplikasi *client*

Aplikasi *client* (Gambar 5), menyediakan pengaturan alamat *serverREST WebService*. Pada aplikasi ini juga disediakan *file-file* yang dapat digunakan untuk proses pengujian. Proses pengujian dikategorikan menjadi dua, yaitu pengujian

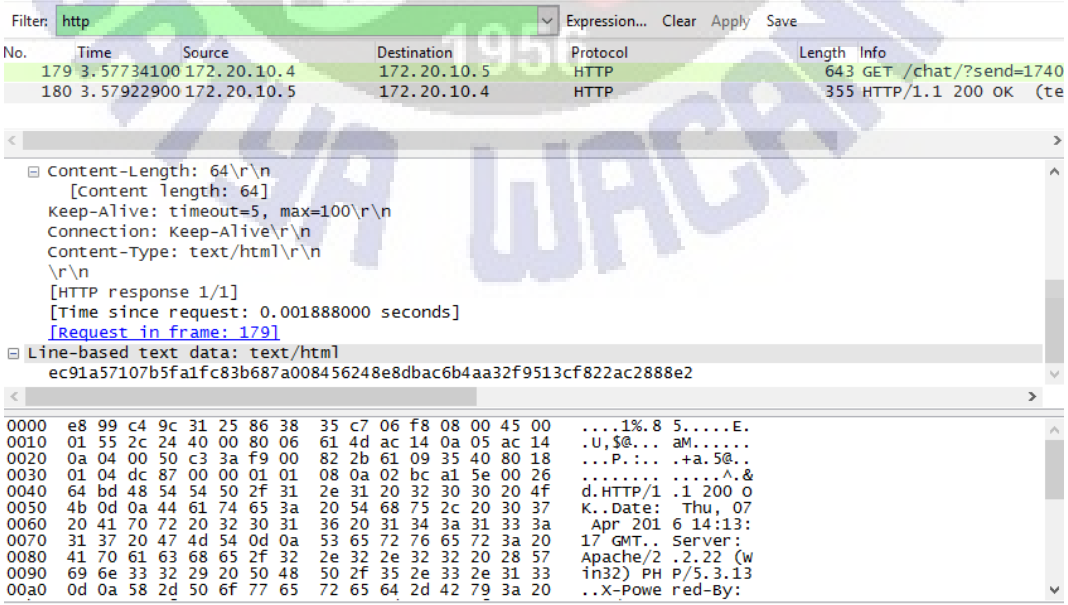
download dan pengujian *upload*. Hasil pengujian berupa catatan waktu enkripsi, catatan waktu pengiriman, catatan ukuran *file* awal dan catatan ukuran *file* akhir.

Pengujian aplikasi ditekankan pada pengujian keamanan. Proses enkripsi terjadi pada dua titik, yaitu ketika pesan dikirim dari aplikasi *client* ke *server*, dan dari *server* ke aplikasi *client*.



Gambar 6 Hasil Analisis Pesan Aplikasi *Client* ke *REST WebService*

Gambar 6 merupakan hasil pengujian dengan aplikasi WireShark. Packet yang dibaca adalah packet yang berasal dari aplikasi *client*yang menuju ke *server*. Perangkat *client*menggunakan ip address 172.20.10.4, dan *Server* menggunakan ip address 172.20.10.5. Ditunjukkan pada bagian Request URI, data yang dikirimkan dalam bentuk *ciphertext*, yang dikonversi ke bentuk heksadesimal.



Gambar 7 Hasil Analisis Pesan dari *REST WebService*ke Aplikasi *Client*

Gambar 7 merupakan hasil pengujian dengan aplikasi WireShark untuk packet yang merupakan balasan dari *server* menuju aplikasi *client*. Data yang berada dalam packet TCP, berada dalam bentuk *ciphertext*.

Kode Program 1 Enkripsi/Dekripsi Algoritma *Vigenere* dengan C#

```
1.  public static byte[] Encrypt(byte[] data, byte[] key)
2.  {
3.          key = ExpandKey(key, data.Length);
4.          List<byte> result = new List<byte>();
5.          int keyIndex = 0;
6.          foreach (byte M in data)
7.          {
8.                  byte C = (byte)((M + key[keyIndex]) % 256);
9.                  result.Add(C);
10.                 keyIndex++;
11.         }
12.
13.         return result.ToArray();
14. }
15.
16. public static byte[] Decrypt(byte[] data, byte[] key)
17. {
18.         key = ExpandKey(key, data.Length);
19.         List<byte> result = new List<byte>();
20.         int keyIndex = 0;
21.         foreach (byte M in data)
22.         {
23.                 byte K = key[keyIndex];
24.                 if (K <= M)
25.                 {
26.                         byte C = (byte)((M - K) % 256);
27.                         result.Add(C);
28.                 }
29.                 else
30.                 {
31.                         int iM = M;
32.                         int iK = K;
33.
34.                         byte C =
35.                                 (byte)((256 + iM - iK) % 256);
36.                         result.Add(C);
37.                 }
38.
39.                 keyIndex++;
40.         }
41.
42.         return result.ToArray();
43. }
```

Kode Program 1 merupakan baris perintah yang digunakan untuk proses enkripsi dan dekripsi, baik pada aplikasi *client*, maupun aplikasi *server*. Proses enkripsi maupun dekripsi bekerja pada level *byte*, sehingga *plaintext* maupun *ciphertext* harus diubah terlebih dahulu ke dalam format byte array, atau dengan kata lain harus dalam bentuk nilai kode ASCII nya.

Pada model yang telah dibuat, dilakukan beberapa pengujian. Pengujian pertama adalah pengujian keutuhan pesan. Pengujian ini bertujuan untuk membuktikan bahwa pesan tidak mengalami perubahan sebelum dan sesudah proses enkripsi dan dekripsi.

Tabel2 Hasil Pengujian Keutuhan Pesan

File Pesan	Ukuran Awal (Bytes)	Nilai Hash Awal	Ukuran *ciphertext* (Bytes)	Ukuran hasil encoding base64 (Bytes)	Ukuran *file* yang dikirim (Bytes)	Ukuran hasil decoding base64 (Bytes)	Ukuran *plaintext* (Bytes)	Nilai Hash Akhir
10	1000	a331487d a5c91016 7bbacc42 d798c307	1000	1334	1334	1000	1000	a331487d a5c91016 7bbacc42 d798c307
50	5000	2d1ddb3d bd34bd9e 633fe6dd ce817e1b	5000	6667	6667	5000	5000	2d1ddb3d bd34bd9e 633fe6dd ce817e1b
100	10000	47a7faa6 ecc55d14 2ea15593 edf90afa	10000	133334	133334	10000	10000	47a7faa6 ecc55d14 2ea15593 edf90afa
500	50000	8e46cbf6 4bc55eb9 1a0120a0 810bb220	50000	666667	666667	50000	50000	8e46cbf6 4bc55eb9 1a0120a0 810bb220
10000	100000	a3dcabc7 b0804282 4c575009 f4abe029	100000	1333334	1333334	100000	100000	a3dcabc7 b0804282 4c575009 f4abe029

Tabel 2 menunjukkan hasil pengujian keutuhan pesan. Berdasarkan hasil pengujian tersebut, maka disimpulkan bahwa, pesan tidak mengalami kerusakan/perubahan dari kondisi awalnya. Perlu diperhatikan bahwa untuk dapat dikirimkan melalui HTTP, data harus dalam bentuk teks, sesuai dengan namanya, *hypertext transfter protocol*. Oleh karena itu *fileplaintext* maupun *ciphertext* harus dikonversi ke dalam format yang murni teks, tanpa karakter-karakter non cetak (spasi, *tab*, *backspace*, *null*, dan lain-lain). Konversi ini dilakukan dengan menggunakan *base64 encoding/decoding*. Proses konversi mengakibatkan perubahan panjang teks. Namun proses enkripsi dan dekripsi sendiri tidak mengakibatkan perubahan panjang data. Hasil perbandingan nilai hash, menunjukkan bahwa tidak ada perubahan data didalam *file* tersebut. Nilai hash diperoleh dengan menggunakan perhitungan algoritma MD5.

Pengujian kedua adalah pengujian pengaruh panjang kunci dan panjang pesan terhadap waktu proses enkripsi/dekripsi. Pengujian ini bertujuan untuk membuktikan apakah terdapat pengaruh antara panjang kunci terhadap waktu proses, dan panjang pesan terhadap waktu proses.

Tabel 3 Hasil Pengujian Pengaruh Panjang Kunci dan Panjang Pesan

File	Panjang Pesan (byte)	Panjang Kunci (byte)	Waktu Enkripsi (milidetik)	Waktu Dekripsi (milidetik)
file1.txt	1000	4	1.02	1.02
file1.txt	1000	8	1.02	1.02
file1.txt	1000	16	1.02	1.02
file1.txt	1000	32	1.02	1.02
file1.txt	1000	64	1.02	1.02

file1.txt	1000	16	1.02	1.02
file2.txt	5000	16	5.11	5.10
file3.txt	10000	16	10.23	10.21
file4.txt	50000	16	21.15	21.05
file5.txt	100000	16	42.30	42.10

Tabel 3 menunjukkan hasil pengujian pengaruh panjang kunci dan pesan. Berdasarkan hasil pengujian tersebut, diketahui bahwa panjang kunci tidak memberikan pengaruh pada waktu proses enkripsi maupun dekripsi. Panjang pesan memberikan pengaruh, dengan perbandingan linier, yaitu, semakin panjang pesan, semakin lama waktu proses.

5. Kesimpulan

Berdasarkan penelitian, pengujian dan analisis terhadap sistem, maka dapat diambil kesimpulan bahwa algoritma *Vigenere* dapat memberikan keamanan pada REST *WebService.* Panjang *ciphertext* tidak mengalami perubahan, karena sifat *Vigenerecipher* yang merupakan stream *cipher* (tanpa adanya *padding*), sehingga tidak ada tambahan beban pengiriman. Panjang kunci tidak mempengaruhi proses enkripsi/dekripsi, karena terdapat dengan panjang kunci berapapun, proses pembentukan kunci dipengaruhi oleh panjang pesan. Pesan yang dikirimkan tidak mengalami kerusakan, dengan arti bahwa proses enkripsi kemudian dekripsi, tidak mengakibatkan perubahan pada informasi pesan. Pada sisi keamanan, pesan yang dikirimkan lewat protokol HTTP, sudah berada dalam bentuk *ciphertext*, sehingga dapat dijamin keamanan dari pesan tersebut, dengan catatan, kunci untuk enkripsi/dekripsi tidak diketahui oleh orang lain.Pada penelitian ini, hanya dianalisis data dalam bentuk teks. Sehingga masih tersisa ruang untuk penelitian selanjutnya untuk menganalisis data multimedia (gambar, audio, video).

6. Daftar Pustaka

[1]. Rodriguez, A. 2015. *RESTful Web services: The basics*. http://www.ibm.com/developerworks/library/ws-restful/. Diakses 10 Mei 2016.

[2]. Kurniawan, E. 2015. *Implementasi REST WebService untuk Sales Order dan Sales Tracking Berbasis Mobile*. Jurnal Eksplorasi Karya Sistem Informasi dan Sains 7.

[3]. Rahman, M. A., Kuswardayan, I. & Hariadi, R. R. 2013. *Perancangan dan Implementasi RESTful Web Service untuk Game Sosial Food Merchant Saga pada perangkat Android*. Teknik Informatika ITS 1.

[4]. Forouzan, B. A. 2007. *Cryptography & Network Security*. McGraw-Hill, Inc.

[5]. Munir, R. 2006. *Kriptografi*. Informatika, Bandung

[6]. Abrihama, D. 2008. *Keystream Vigenere Cipher: Modifikasi Vigenere Cipher dengan Pendekatan Keystream Generator*. Program Studi

Informatika ITB. Bandung
[7]. Salomon, D. 2005. *Coding for data and computer communications*. (doi:10.1007/b102531)
[8]. Hamad, H., Saad, M. & Abed, R. 2010. *Performance Evaluation of RESTful Web Services for Mobile Devices.* Int. Arab J. e-Technol. 1, 72–78.
[9]. Wagh, K. & Thool, R. 2012. *A comparative study of soap vs REST WebServices provisioning techniques for mobile host*. Journal of Information Engineering and Applications 2, 12–16.
[10]. Tutorials Point 2016. *SDLC - Waterfall Model*. https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm. Diakses pada 2 Agustus 2016.